CREODIAS Deployer

User Manual

Introduction

Welcome to the CREODIAS Deployer User Manual. This document provides a comprehensive guide to using the CREODIAS Deployer, an integral part of the Eclipse Graphene project.

Prerequisites

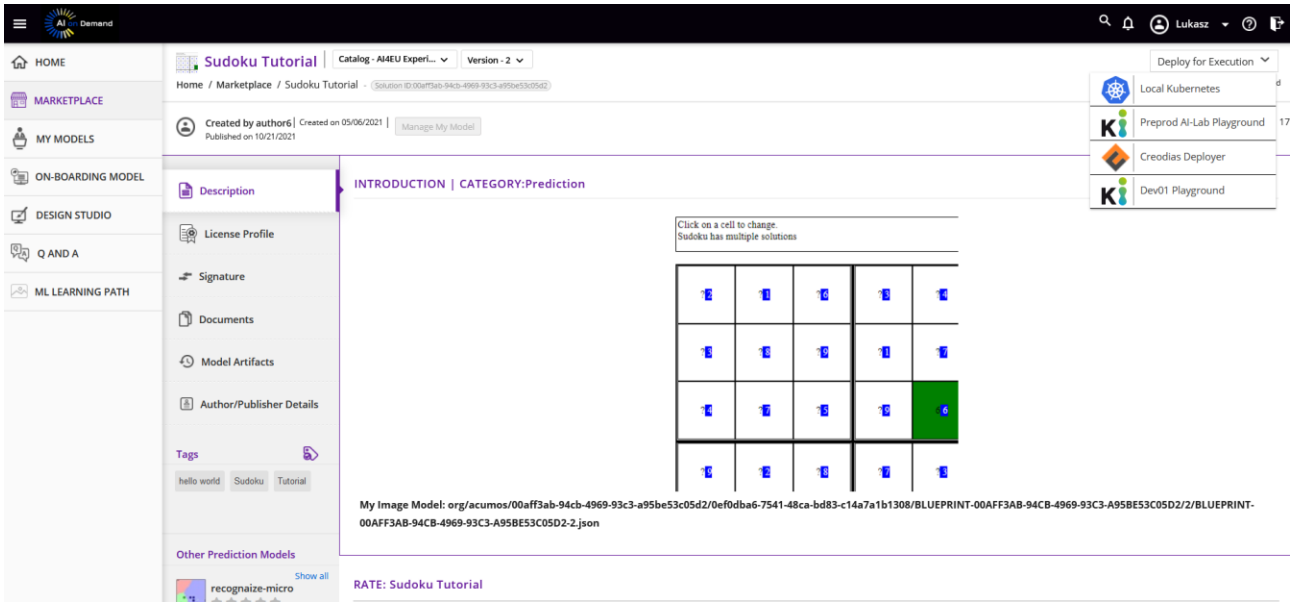
Before you begin, it's essential to understand that the CREODIAS Deployer facilitates the deployment of experiments from the AI on Demand portal to Kubernetes clusters hosted on the CREODIAS platform. To ensure smooth deployments, please ensure the following:

- A Kubernetes cluster has been set up on the CREODIAS platform. Detailed instructions can be found in the "How to Create a Kubernetes Cluster Using Creodias OpenStack Magnum" section of the CREODIAS FAQ.
- The Kubernetes cluster should be publicly accessible over the Internet. This is vital as AI on Demand and CREODIAS operate in distinct execution environments.

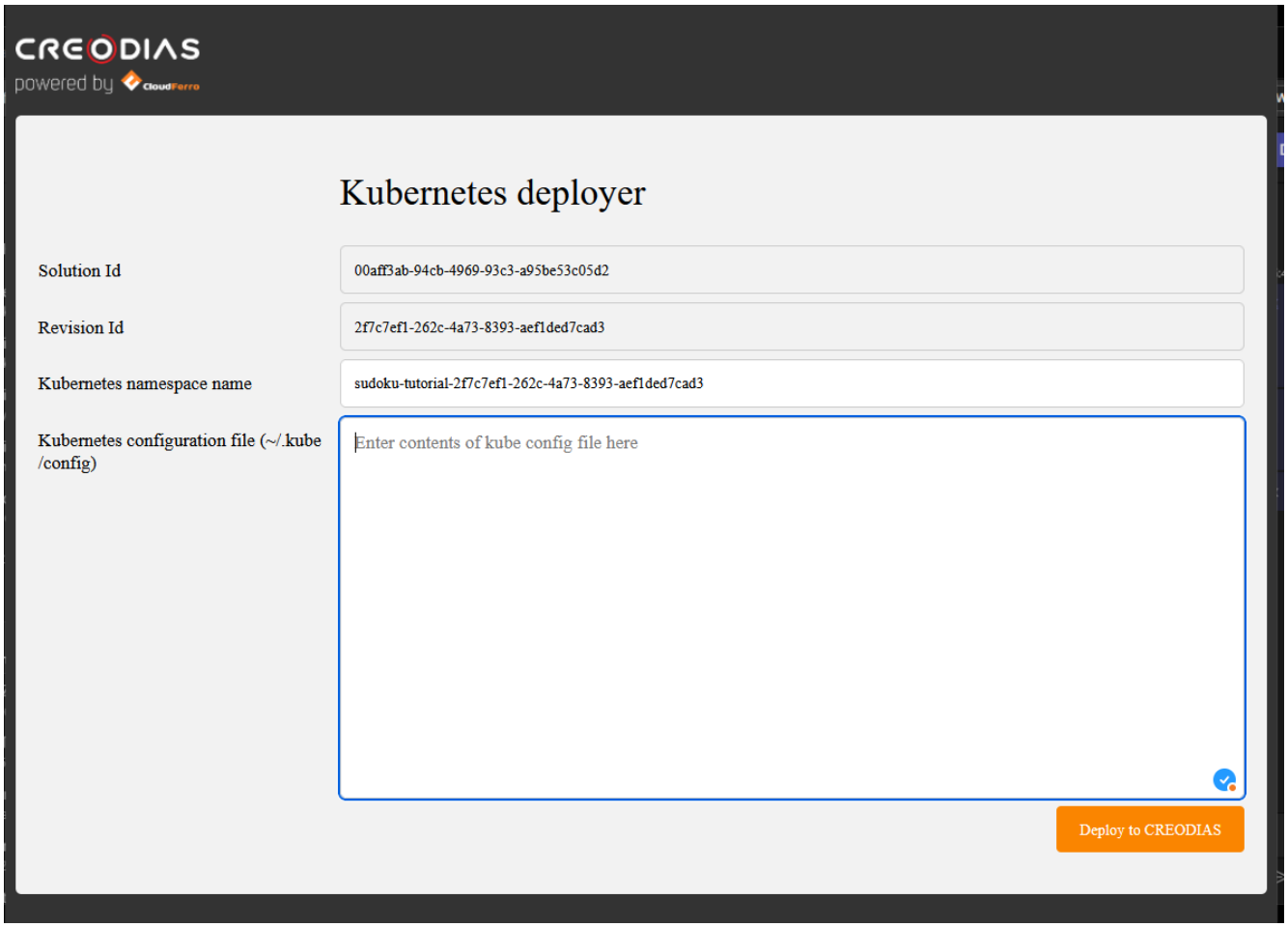
Deploying an Experiment

Follow the steps below to deploy the Sudoku Tutorial experiment available on AI on Demand:

1. **Navigate to the Sudoku Tutorial Experiment Page**:
    - Choose the "Creodias Deployer" option.

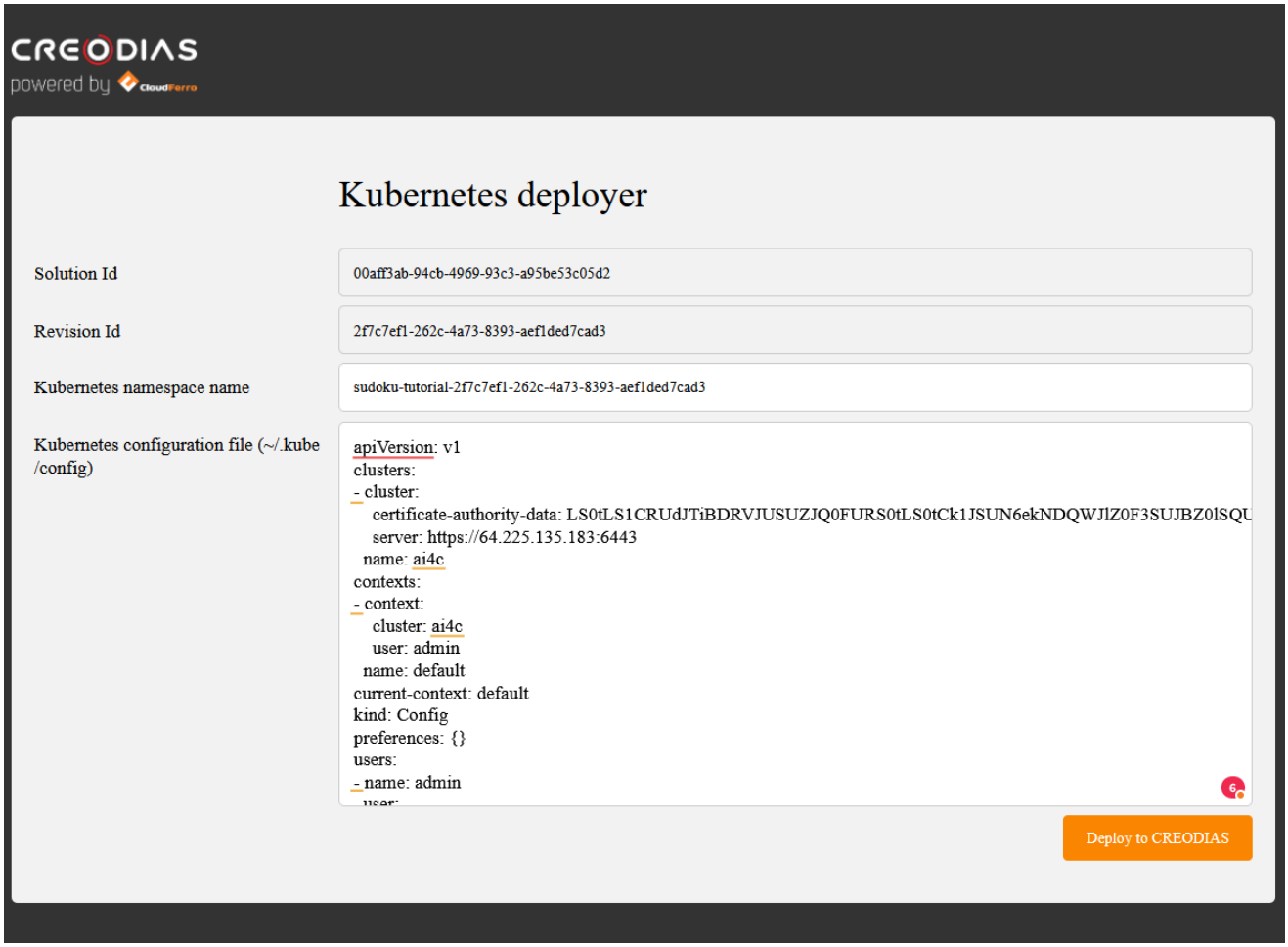


2. **Access the Deployment Page**:
    - Upon selection, the deployment page will appear.



- The system will generate a default namespace for your Kubernetes deployment, combining both the solution ID and revision ID. If necessary, you can modify this namespace.

3. **Provide the Necessary Configuration**:
    - Enter your Kubernetes configuration file for the target cluster established in CREODIAS.



4. **Initiate Deployment**:
    - After providing the Kubernetes configuration and defining your namespace, click the 'Deploy to CREODIAS' button to start the deployment process.

5. **Monitoring Deployment Progress**:
    - The deployer will begin the process by creating the necessary components such as namespaces, services, deployments, persistent volume claims, etc., for setting up your solution.
    - Depending on the complexity of your solution, it could be simple (single container) or composite (featuring an embedded pipeline blueprint). For composite solutions, they will be automatically initiated once components have been successfully deployed and are operational.

6. **Error Handling**:
    - If you encounter errors during deployment, you can retry. The deployer is designed to identify which tasks were successfully executed, allowing it to skip them and continue the deployment seamlessly.

7. **Completion & Summary**:
    - Upon successful deployment, a summary page will be displayed, offering links to all deployed services.